Deploying and Debugging Job Runner

This document explains how to deploy and debug Job Runner.

# Installing the Job Runner Service

Job Runner is designed to run as a Windows Service that continuously runs and checks for any jobs that need to execute at a certain point in time. This section explains how to install the Job Runner Service on a client machine.

## Create Job Service Event Log

The Job Runner Service requires a Windows Event Log be installed on the client machine that is running the service. It uses this event log to write information when debugging. The following steps explain how to create the event log:

1. **Open** the **JobRunner solution** in Visual Studio.
2. **Rebuild** the **JobServiceEventLogCreator** project using the desired solution configuration.

*Note:* The solution configuration determines what the name of the log will be. By default the name of the log is called “JobsServiceLog”. However, different configurations can be created for different instances of the JobsService. For example, there is a configuration called “Gensets”, which is a specific instance of the JobsService. Therefore, by selecting the Gensets solution configuration the log name will match the value specified in the App.config specific to that solution configuration. If you look in the App.Gensets.config file, you will notice that there is a configuration section called “jobsservice” with a child node called “log”. The Log xml node has a “name” attribute which is the value used when creating the name of the log.

1. Navigate to the **bin\{Solution Configuration}** folder for this project in Windows Explorer.
2. **Copy** the **contents** of this folder **to the client machine** that the service is going to be running on.
3. On the client machine, **execute** the **JobServiceEventLogCreator.exe**. This will create the event log. You can verify that the event log is created by going to Windows Event Viewer and expanding the “Applications and Services Log” section. The log name will match the name specified in the *App.{Solution Configuration}.config* file.

## Install Job Runner Service

The following steps explain how to create the Job Runner Service:

1. **Open** the **JobRunner solution** in Visual Studio.
2. **Rebuild** the **JobsService** project using the desired solution configuration.

*Note:* The solution configuration determines the valid plugin paths, event log information, and email configuration for unhandled exceptions. The specific *App.{Solution Configuration}.config* for each solution configuration are used to control different settings based on the selected configuration.

1. Navigate to the **bin\{Solution Configuration}** folder for this project in Windows Explorer.
2. **Copy** the **contents** of this folder to the desired location on the **client machine**.
3. Install the windows service on the client machine using the instructions provided by Microsoft. [How to: Install and Uninstall Services](https://msdn.microsoft.com/en-us/library/sd8zc8ha(v=vs.110).aspx)

*Note:* When installing the service it will ask for a username and password. Provide the username and password for the user account that you would like the service to run as. Beware that the domain is needed along with the username.

1. On the client machine, **open** the **Services** tool and verify that the service is in the list.

## Starting the Job Runner Service

To start the Job Runner Service, open the **Services** tool on the client machine that has the service installed. Locate the service that you would like to start and click the “Start” link that shows up in the left pane or right-click menu. This will start the service without debugging turned on. If you would like to start the service with debugging turned on, open the **Properties** window for the service (right-click > Properties) and in the **Start parameters** textbox enter **debug**, then click the **Start** button. This will start the service and log information to the event log specified in the *JobsService.exe.Config* file that is local to the *JobsService.exe* that the service is executing.

## Current Job Service Instances

Currently there are two instances of the Job Runner Service, the Default instance and the Gensets instance. The instance is determined by the solution configuration that is selected. The Gensets instance is used when the “Gensets” solution configuration is selected and the Default instance is used for any other configuration.

# Deploying Job Runner Jobs

Job Runner utilizes Managed Extensibility Framework (MEF) to provide a plugin like environment where jobs can be added and removed on demand. It uses the settings defined in the App.config under the *JobRunner > pluginpaths* section which defines the folders to look in for any classes that are decorated with the *System.ComponentModel.Composition.ExportAttribute* with the type of *JobRunner.IJob*. This section explains how to deploy the jobs so the service can run them:

1. **Stop** the **Job Runner Service** on the client machine where you will be deploying jobs.
2. **Open** the **solution** in Visual Studio that contains the Job Runner jobs that you would like to deploy.
3. **Rebuild** the project that contains the jobs using the desired solution configuration.
4. Navigate to the **bin\{Solution Configuration}** folder for this project in Windows Explorer.
5. On the client machine, **navigate** to the location of the **EXE** that the Job Runner Service is running. From there, navigate to the destination folder that is designated for the jobs of the application you are working with.

*Note:* Currently, we have a convention established where there is a **plugins** folder in the same folder as the **EXE** that the Job Runner Service is running. Inside of the **plugins** folder, there is a subfolder that is specific to each application that relies on Job Runner for executing jobs. The specific jobs for each application should be placed in the corresponding folder for that application. For example, mpora jobs should be placed in the *plugins\mpora* folder.

1. If the application specific plugin folder does not exist you can create it, however this new path will need to be added to the *JobRunner > pluginpaths* section of the **App.{Solution Configuration}.config** file in the **JobRunner.JobsService** project. Then the JobsService project should be rebuilt and redeployed back out to the client machine so the new configuration settings exist.
2. **Copy** the **dll** files and the **{Application Jobs Assembly}.dll.config** file from the **bin\{Solution Configuration}** folder on your local machine to the destination folder on the client machine from the previous step.

*Note:* If there are **dll** files that are already included in the folder that the Job Runner Service **EXE** is running out of, they do not need to be included here. Examples:

* Encompass.Concepts.dll
* Encompass.Simple.dll
* JobRunner.dll

1. **Start** the **Job Runner Service**.

# Debugging Using JobsServiceDebugger

The *JobRunner.JobsServiceDebugger* project facilitates in testing and debugging Job Runner. It is intended to simulate the JobsService while allowing for debugging in Visual Studio. In order to use the JobsServiceDebugger, the Job Service Event Log must exist on your local machine. Please refer to [Create Job Service Event Log](#_Create_Job_Service) section of this document for instructions. The following steps explain how to use the JobsServiceDebugger:

1. **Open** the **solution** in Visual Studio that contains the Job Runner jobs that you would like to debug.
2. If the JobsServiceDebugger project is not already added to the solution, **Add** the **JobsServiceDebugger** by right-clicking the Solution in Solution Explorer, clicking Add > Existing Project, then navigating to the location of the JobRunner solution on your local machine and selecting the JobServiceDebugger.csproj file.
3. **Rebuild** the **JobsServiceDebugger** project using the desired solution configuration.
4. Navigate to the **JobsServiceDebugger\bin\{Solution Configuration}** folder and **create** a **plugins** folder with a subfolder that matches the application you are debugging.

*Note:* The plugin path must match of one of the plugin paths defined in the **App.{Solution Configuration}.config** file in the JobServiceDebugger project.

1. **Rebuild** the project that contains the jobs you would like to debug using the desired solution configuration.
2. Navigate to the **bin\{Solution Configuration}** folder for this project in Windows Explorer.
3. **Copy** the **dll** files and the **{Application Jobs Assembly}.dll.config** file from the **bin\{Solution Configuration}** folder to the **JobsServiceDebugger\bin\plugin\{Application}** folder from the previous steps.
4. Set the **JobsServiceDebugger** as the **Startup Project**.
5. Click the **Start** button in Visual Studio to start debugging. Any breakpoints within the job code should now be reachable.

*Note:* **Steps 5-7 need to be repeated every time a change is made to the job code!**